

# viSQLizer: Using visualization for learning SQL

Kristin Annabel Torjussen Folland

Department of Computer Science and Media Technology,  
Norwegian University of Science and Technology

## Abstract

Structured Query Language (SQL) is used for interaction between database technology and its users. In higher education, students often struggle with understanding the underlying logic of SQL, thus have trouble with understanding how and why a result table is created from a query. A prototype of a visual learning tool for SQL, viSQLizer, has been developed to determine if visualizations could help students create a mental model and thus enhance their understanding of the underlying logic of SQL. Through the use of animations and decomposing, our results indicate that visualizations might give students a better understanding of the underlying logic, and that students gain the same learning outcome through visualizations as when using an online tutorial with explanatory text and exercises. Feedback from both professors and students from conducted interviews and experiments indicate that the tool could be used by professors as a visualization tool in lectures, and by students as a practical tool; not as a replacement of, but as an addition to traditional teaching methods.

## 1 Introduction

Database Systems is an important subject in Computer Science degrees. Data manipulation and organization have been pillars in Computer Science degrees for decades [1, p. 154]. The learning methodologies used today by course coordinators varies, but often consist of lectures, lab sessions and programming projects, and advanced DBMS systems or blackboards are often used to visualize the database structure and logic [2, 3].

Different database professors organize database courses differently, and assess the students on different topics, skills and knowledge. DBMS concepts, database design and transaction processing are all topics covered in most database courses [2]. In the course "*Database Systems*" at Colorado State University, "*Query languages*" and "*Relational databases*" are the topics that are allocated most time within a duration of fifteen weeks [2, p. 299]. Query languages are perceived as a difficult subject for the students to learn, and students are expected to know how to create queries when they graduate with a Computer Science degree [4].

Structured Query Language (SQL) is supported by relational database systems and is a seemingly simple way to access and change data in relational databases [5], but students

---

*This article was presented at the UDIT / NIK-2016 conference; see <http://www.nik.no/>.*

often struggle with learning the fundamental SQL concepts and the syntax. According to Sadiq et al. [6], the SQL SELECT command is often perceived by students as simple but is in fact capable of formulating complicated queries. Kearns et al. states that "*students do not understand the fundamental concepts that drive the language*" [4], and tend to miss "*the linkage clause from a multi-table query*" [4] or make other syntax errors, resulting in a result table with incorrect data. Several researchers state that students have difficulties learning SQL [6, 7, 8].

To solve this problem, this study will investigate the use of visualizations in an interactive learning tool for teaching novice students SQL. A learning tool could give students a better understanding of SQL, by visualizing the underlying logic and letting students learn through a practical approach. This paper is based on a Master Thesis, written at NTNU Gjøvik [9].

## **Related work**

Several online SQL tutorials are available today, which students can use free of charge to help them learn SQL syntax [10, 11, 12]. These use table presentation as well as informative text to teach SQL to their users. The learning effects of this type of e-learning has been researched, and the results have been positive [13], but we think that this might not give the students the understanding of the underlying logic that is needed to write good, understandable SQL queries, because this type of learning does not give students information about how the syntax actually work in practice "behind the scenes".

Some have used game elements to facilitate learning of SQL [14, 15, 16]. Serious games provide students with a new way of learning, which might both be fun and motivating. Game elements in a SQL learning tool could, however, make SQL too abstract; different from the real world scenario, and might not give students the understanding that they need to master SQL.

A few learning tools use animations to facilitate learning [7, 17]. Animations and visualizations have often been used in various disciplines to support complex topics [18], and research has shown that students "*respond well to a practice-in-theory approach*" [19]. Allenstein et al. proposes that visualization can be used as a tool to provide "*another way to think about a given concept*" [19]. According to Patwardhan & Murthy [20], interactive visualization offers a considerable amount of benefits for students "*such as explanation ability, comprehension, ability to correlate scientific concepts, learning of conceptual and procedural knowledge, process of skill acquisition, building of mental models, as well as increased learners' engagement*" [20].

According to Prior [8], students have problems with mentally visualizing the intermediary result from the execution of various SQL operations. SQL is a declarative language, which means that the query specifies what is to be done in the database, whereas a procedural language explains how to do it. As such, SQL queries cannot be separated into various steps to be solved separately, but can instead be separated into temporary sets of data, resulting from various parts of the same query. This can be done through decomposition, where the query is broken down into smaller, simpler queries/sets. This is done by separating the various operations in the query through a parser, and representing them in their natural order [7].

Kearns et al. [4] and Danaparamita & Gatterbauer [21] have developed SQL learning tools using decomposition, and Cembalo et al. [7] uses both decomposition and animation in their SQL learning tool. There has, however, not been done any experiments on the effectiveness of the learning activity when visualizing with animation and

decomposition. We want to determine if temporal sets using decomposition could help students understand the underlying logic of SQL and if visualization in a learning tool using animations could make the learning activity more effective.

## **Contributions**

The first contribution will be to develop a prototype of a visual SQL learning tool using decomposition and animations. This prototype will be used in a student experiment, where we want to look at if this type of learning tool could help students create a mental model, and thus an understanding of the underlying logic of SQL. Secondly, we want to determine if students find the learning tool motivating and/or engaging. Third, we want to determine students' learning outcome per time unit when using a visual learning tool contra an online SQL tutorial. And finally, we want to look at how such a learning tool could be applied in relation to traditional methods for teaching SQL.

This paper is organized as follows. Section 2 presents the different methods used. Section 3 presents the learning tool prototype, viSQLizer. Section 4 our results, while Section 5 presents our conclusion and a few suggestions for future work.

## **2 Methodology**

The project work was structured to follow a path of different methods, as shown in Figure 1. The goal was to develop a prototype learning tool that could give students an understanding of SQL logics, in accordance with experts wants and needs in the SQL field, and students' experience of different visualizations of SQL operations.

To determine what topics should be supported in the learning tool, expert interviews with three database professors from different campuses at NTNU were conducted. The interview questions were related to students' threshold concepts, the database professors' teaching methods, and the database professors' need of a SQL learning tool. This information helped us decide what to focus on during the development of our viSQLizer prototype.

The development process was done in two iterations, with a preliminary user test between the two, conducted on five second year students from the Game Development track at NTNU Gjøvik. Through this user test we got information about how the students perceived the various visualization ideas implemented in the viSQLizer prototype. This information helped us decide which visualization ideas to keep in the next development iteration.

When the development phase was done, an experiment was conducted to measure the students' understanding of the underlying logic of SQL. This could give us some indications of whether the visualizations implemented in the viSQLizer prototype helps students understand SQL on a higher level. The experiment was conducted on 17 first year Game Development and Software Engineering bachelor students at NTNU Gjøvik. We wanted to conduct the experiment with development students who had not been enrolled in a database course yet but who are to follow such a course the following semester. These students could give us a good overview of the learning outcome after a given time unit, since they have no previous experience with SQL.

The experiment followed the 2-group post-test experimental design. No pre-test was conducted because the students in our sample had none or limited experience with SQL. The students participating in the experiment were randomly divided into two groups, a control group and an experimental group, and placed in two different rooms. The two

groups were to go through a number of tasks in their respective learning tool, an online learning tool, Vertabelo Academy [10], or our viSQLizer prototype <sup>2</sup>. The experiment setup was as follows:

1. All participating students watched a short introductory video, explaining databases and SQL, created by Rune Hjelsvold, the responsible professor in the database course for the second year bachelor students at NTNU Gjøvik.
2. Each student in each of the two groups were given a set of exercises they were supposed to work with in their respective learning tool (viSQLizer or Vertabelo Academy [10]).
3. All participating students were given the same test questions on paper, and were asked to answer these without any aid.
4. 5 students from each group were picked out randomly to participate in a short interview about the learning tools and the experiment setup.

Feedback was given to us from database professors on the final version of the prototype, in terms of if and how they would like to use our developed learning tool in their database courses.

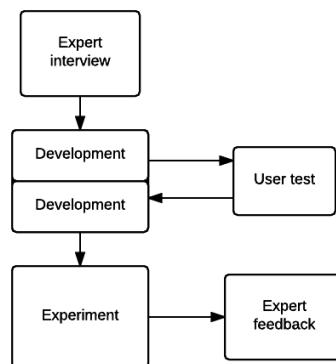


Figure 1: The project structure

### 3 viSQLizer

The developed learning tool prototype, named viSQLizer, was developed using PHP 5.6.2 with MySQLi connection to the MySQL databases, and HTML5, which makes it viewable in all modern browsers. The animations was created using HTML5 Canvas and the suite of modular libraries, CreateJS<sup>1</sup> with EaselJS and TweenJS, and the decomposition was developed using the PHP SQL parser PHP-SQL-Parser [22]. The solution contains two databases. One for storing information about the saved queries, and one for the available data sets. User restrictions are set so that a student cannot delete or change the provided sample data. The viSQLizer learning tool is built on the Decomposer project [23].

Results from the expert interviews indicated that one of the students' threshold concepts are JOIN, that the database professors use labs and a blackboard when teaching SQL, and that the three database professors would like to use a SQL learning tool as part of their lectures or as an offer to the students as an addition to the traditional lectures.

---

<sup>1</sup><http://www.createjs.com>

## Decomposition

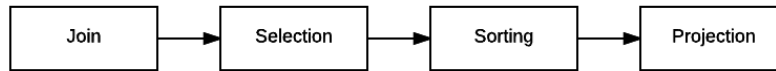


Figure 2: Order of the algebra operations

The final version of the viSQLizer prototype decomposes the submitted query into multiple steps. Each step represents one individual part of the query. The different steps contains algebra operations represented in the order as shown in Figure 2. To explain the decomposition in detail, an example query is provided; *"SELECT occupation, start\_date FROM work WHERE occupation = 'lawyer'"*. This query is decomposed into two steps, which are explained below.

1. An **asterisk (\*)** replaces the column names provided in the original query. The rows that fulfills the **WHERE** requirements are displayed in the result table. The query for this step is; *"SELECT \* FROM work WHERE occupation = 'lawyer'"*.
2. The columns *occupation* and *start\_date* are **selected** from the original *work* table in the database. The query for this step is; *"SELECT occupation, start\_date FROM work WHERE occupation = 'lawyer'"*.

## Design

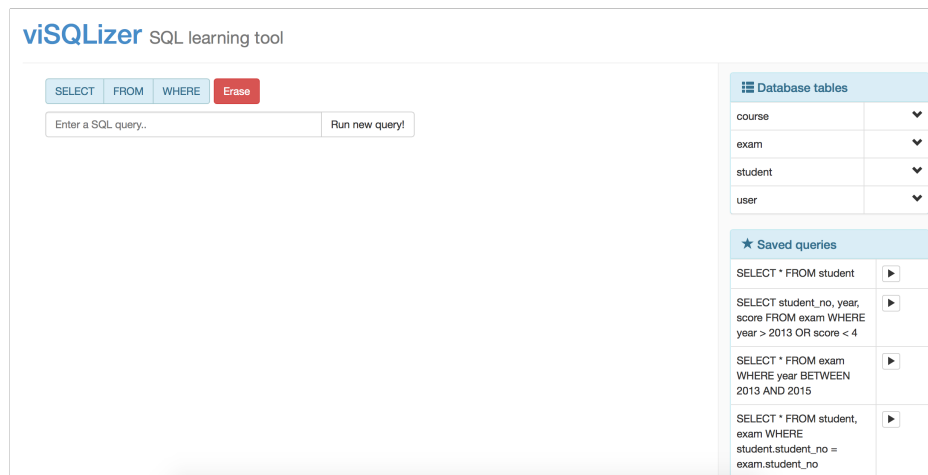


Figure 3: The front-page of viSQLizer

The final design of viSQLizer is shown in Figure 3. The different elements in the view will be shortly described below.

A table is available to the students using viSQLizer, containing the information about the available database tables and column names. This was implemented to minimize the students' cognitive load, so they they do not have to remember this information when using the prototype. This table can be seen as "Database tables" in Figure 3. Another table is also present in the viSQLizer prototype, the "Saved queries" table; queries that can be submitted, decomposed and visualized. These queries are various example queries that are supported by the prototype.

When a query is submitted in the viSQLizer prototype, it gets decomposed into multiple steps. The number of steps that the decomposer creates are displayed, and the part of the query currently visualized is displayed in a blue information box. New text appearing in this information box throughout the decomposed steps is marked in bold, so that the student easily understands what is being visualized in the current step.

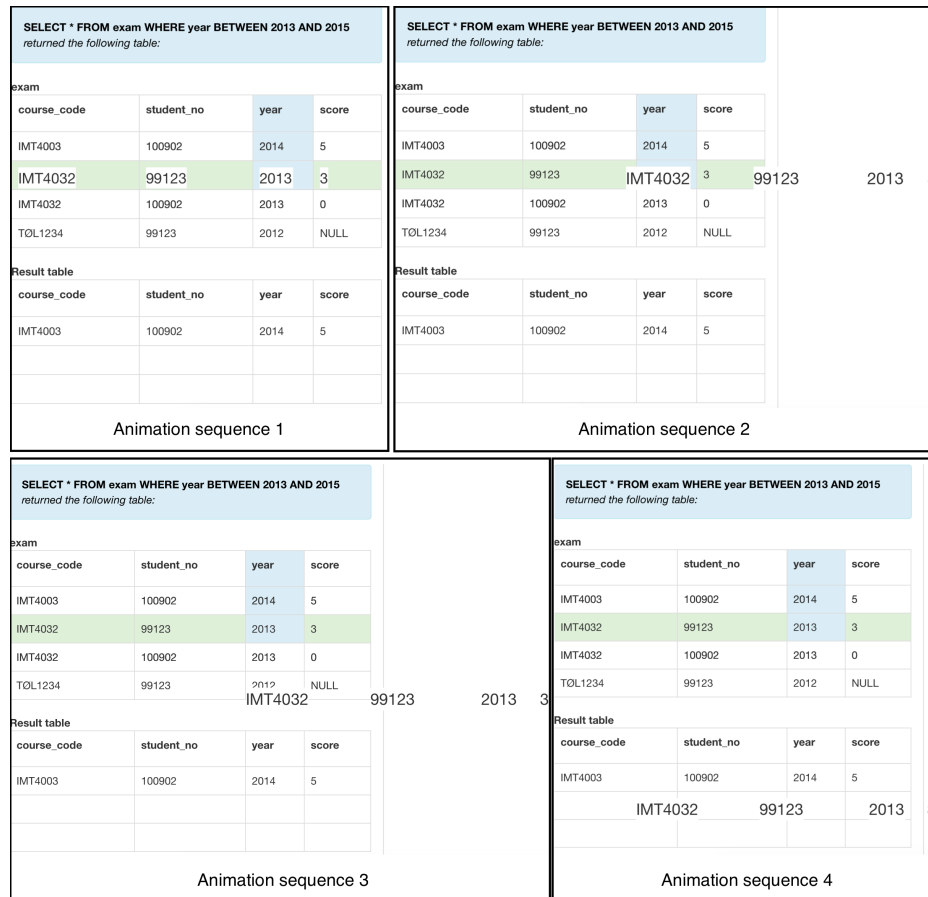


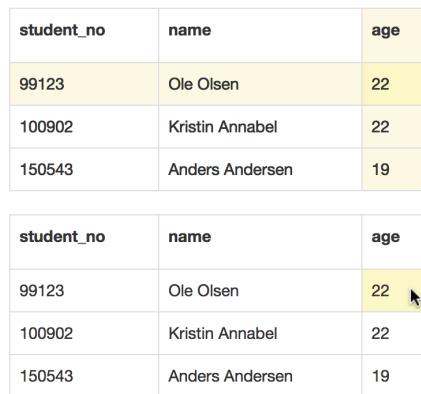
Figure 4: Animating one row from the original table to the result table

When a query is decomposed in the solution, two tables are represented to the student. In the first step, e.g. as shown in Figure 4, the table displayed at the top is the original table, as presented in the database. In Figure 4, this is the exam-table in the sample database. In the next steps, the table displayed at the top will contain the result table from the previous step. This is done because the next steps in the decomposition relies on the data from the previous part of the query.

The table displayed at the bottom is the result table. This table will contain the data after the animation for the current step has completed. When the animation starts, each cell that corresponds with the active part of the decomposed query in the original table is animated down to the correct position in the result table. The animation sequence is shown in Figure 4. When the animation is completed, the result table will contain the correct data represented by the query.

Colour highlighting has been implemented in the solution to emphasise the animated cells and the ON/WHERE/ORDER BY clauses in the decomposed query. This can be seen in Figure 4. The animated cells get a green highlighting when they are being animated, and the cells affected by the ON/WHERE/ORDER BY clauses are highlighted

in blue in the original table. It is also possible to see the cell origin after the animation has completed, by hovering over any cell in the result table. This can be seen in Figure 5.



student_no	name	age
99123	Ole Olsen	22
100902	Kristin Annabel	22
150543	Anders Andersen	19

student_no	name	age
99123	Ole Olsen	22
100902	Kristin Annabel	22
150543	Anders Andersen	19

Figure 5: Highlighting on the cell, row and column

For more information about the viSQLizer learning tool, please see the Master Thesis [9] or visit the online prototype<sup>2</sup>.

## 4 Results

The main purpose of the student experiment was to compare the learning outcome in a group of students using an online tutorial, Vertabelo Academy [10], with another group of students using the viSQLizer prototype, per time unit. Vertabelo Academy is different from the viSQLizer prototype; it does not use decomposition or animations, and it provides the users with explanatory text and exercises as well as a structured set of tutorial topics with increasing complexity. Sub goals of the experiment was to see if the use of a learning tool motivates the students to learn, and if and how they want to apply the use of such a tool to traditional methods for learning SQL.

As mentioned earlier, the participating students were divided into two groups; One group was the control group, a group of eight students using Vertabelo Academy [10], and the other group was the experimental group, a group of nine students using the viSQLizer prototype<sup>2</sup>.

### Mental model and understanding of underlying logic

When asked about negative aspects when using the learning tool, some students in the control group answered that it was hard to find information about the database, as column names. The students did not mention any other negative aspects. Some students in the experimental group stated that there were too small data sets in the sample database. *"If there are only four rows in a table, you cannot really see what the query is doing"*.

Some of the students in the experimental group mentioned some things that are specific for visualization in the viSQLizer prototype. A student from the experimental group stated *"By using such a tool, it is easy to understand how SQL works"*. Another student from the same group seemed satisfied with how the visualization was organized. *"When you wrote a query, you got all the information you had asked for. And the visualization told you how much data was going to be fetched, before the animation began, by showing the empty result table with the correct number of rows and columns"*.

---

<sup>2</sup>[andmark.no/kristin](http://andmark.no/kristin)

Students in the control group, using Vertabelo Academy [10], were mostly interested in answering correct on the exercises provided in the tutorial. None of the students gave us any indications that they were pondering over how the result table was constructed. Students in the experimental group mentioned that it was easy to understand how SQL work, and commented on elements like small data sets being an issue. We think this indicates that the students understood some of the underlying logic, because they commented on the data flow of table data. The experimental group understood the how and why, and commented on issues that might only be clear for students who understand some of the underlying logic.

## **Motivation**

The students mentioned that they were motivated, interested, and engaged in the learning process, and that this type of learning was quicker than traditional methods for learning SQL. *"I learn quicker, than by reading an entire chapter in a book without understanding how the syntax works". "I was very engaged, and I felt that I did learn better".*

## **Learning outcome**

The students were asked what they liked about using a learning tool, like the one they had tried during the experiment. This question was related to learning tools in general, and the answers from the students were not grouped when analysing the results. Most students stated that they learned quicker by using the learning tool and the practical approach. *"I am learning quicker, and remember it better". "You get a practical approach to the subject, you get to practice and repeat, and then the knowledge sticks better". "By doing a practical approach, you get it into your fingers, better than by hearing about the theory; when you know the theory but not how to apply it".*

The tests were graded with a A-F grading style. The grading was done in cooperation with Rune Hjelsvold. The results from the test conducted in the experiment shows that the students in both groups have done a really good job. The control group, using Vertabelo Academy [10], got an average of 89% correct answers on the test, and the experimental group, using the viSQLizer prototype, got an average of 90% correct answers. This is shown in Figure 6. A confidence interval of 95% is added to the graph. This shows that the average student is getting 86-92% correct answers, which is a B to A grade. The grade span is from A to C, and there are no students who got a lower grade than C in any of the groups.

## **Exercises and explanatory text**

The results indicate that the control group, using Vertabelo Academy [10], liked the use of exercises in the tutorial and the text explaining the various operations in SQL. *"It is essential. It was easy to read. And it was nice to get a confirmation that you had understood everything". "It is a good method for learning. Doing practical exercises really helps the memory".* Students mentioned that the exercises was a good contribution to learning, and that it followed a meaningful path where each exercise is building on the knowledge from the previous exercise. *"It is nice, because it is piecewise learning, and you can decide your own pace".* One student in the experimental group stated that he would like to have textual information in addition to the provided visualization. *"I would like the visualizations to be explained, not just visualized. The current scenario could be explained shortly in text".*



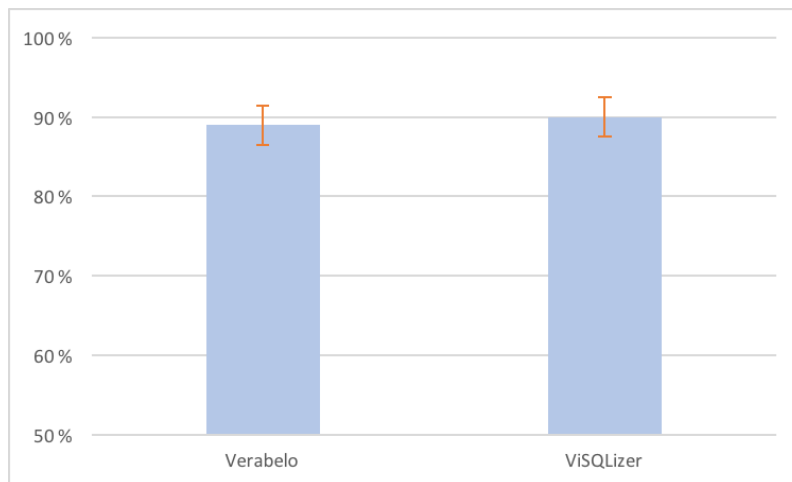


Figure 6: Average percentage value of the results from the two groups

## Usage of the learning tool

One student in the control group stated that he would like to use the learning tool in the database course. *"I hope we get to use a learning tool like this one ourselves, when we are taking the database course next semester"*. One student from the experimental group answered *"... Some of the underlying theory are missing; what actually happens and why. This learning tool could be a supplement to lectures. In lectures, we get more theory behind the practical approach"*.

The results indicate that the students would like to have lectures in addition to using the learning tool so that they have the opportunity to ask a professor for help if they need to. The students stated that they wanted to learn the SQL basics in lectures and use the learning tool for a hands-on approach to learn how SQL works in practice.

The three interviewed database professors were asked to view the prototype, and give some feedback on if/how they would like to use the learning tool in their database lectures. The feedback from the professors was positive. *"I think this learning tool is good and instructive. It has a good explanation of basic SQL"*. *"This is exiting, and it is a tool that I would like to use for teaching"*. *"Testing the learning tool was fun"*. One of the professors stated that the prototype would be great for learning and cramming SQL syntax. *"Furthermore, I think the animation is good as an illustration, and from a pedagogical perspective. This is a tool with great teaching potential as I see it"*. Another professor stated that the learning tool would be an appropriate tool for novice students learning SQL. *"It is a useful tool in the beginning of learning SQL"*.

## 5 Conclusion

Our results indicate that a learning tool with visualizations may help students create a mental model, and thus understanding the underlying logic of SQL. This is based on the fact that these students comment on the data flow in SQL, and not just on the visualization itself. Further research needs to be conducted to determine how much of an impact visualizations in a learning tool could have on the students' understanding of the underlying logic.

Results from the conducted experiment indicates that students think of the viSQLizer prototype as motivating and engaging. Students have stated that they learn more and are more motivated to learn through this learning tool than through traditional lectures and

textbooks. In our opinion, more research needs to be conducted to determine the effects on students' motivation when using a visual learning tool.

When comparing the students' learning outcome when using a visual learning tool contra using an online tutorial, we found that the learning outcome were similar for both groups per time unit. This indicates that visualizations in an SQL learning tool could contribute to a better learning outcome if certain elements from an online tutorial are implemented in the visual learning tool, such as explanatory text and exercises. We suggest that research should be done to determine if these types of elements could improve the students' learning outcome when using a visual learning tool. We also suggest more research to be done to determine the long term effects on students' learning outcome when using a visual learning tool.

Interviews with both students and database professors have shown that a visual learning tool such as the viSQLizer prototype could both be used by professors in lectures, to visualize database tables and the data flow, and by students as a practical addition to traditional teaching methods such as lectures.

Feedback from database professors have been positive towards the prototype, and the database professors have shown real interest in using the tool in their own database course. Students have shown great engagement and seems to be motivated by the viSQLizer learning tool. Students stated that they were interested in using the learning tool in a database course.

## **Discussion**

The viSQLizer learning tool presented in this paper is a prototype developed for the purpose of an experiment of students' understanding of the underlying logic of SQL through visualizations in the learning tool. The prototype is not a full functional tool ready to be used by students/in lectures, and has both bugs and shortcomings. Because this prototype was developed during the duration of a master thesis, we chose to emphasize the SQL operations that students often learn in the first couple of weeks in a SQL course, as JOIN, ORDER BY and WHERE. We also chose not to spend any time fine tuning the prototype, because it was only meant to be used in the student experiment, and thus does not need to have any more functionality than needed in that particular experiment. The number of participating students in the preliminary test and the experiment is a limitation for this study, but we believe that our results still give us an indication of how a visual learning tool could give students some understanding of the underlying logic of SQL.

## **Future work**

The viSQLizer prototype was developed as a learning tool for novice students learning SQL. As a future development, the learning tool could facilitate a higher level of learning by implementing more advanced SQL functions and operations. We will shortly discuss the different possible functions and operations that could be added, and the issues related to the implementation of these functions and operations in viSQLizer as it is implemented today.

Aggregation functions are difficult to animate, because the result cell for the aggregation function is dependent on the original table data and is not present in the original table itself. Thus, the cells for the aggregation functions cannot be animated from the original table to the result table as the animation is implemented in the current version of viSQLizer. This could be solved by inserting the aggregation cells' values after all the other cells in the table has gotten its values.

Sub-queries are not implemented in the viSQLizer prototype, because visualization of sub-queries need multiple tables to be represented. We suggest that sub-queries could be implemented in the prototype by visualizing each part of the sub-query separately and comparing the two queries to get the result table. The design of the prototype has to be modified so that there are room for multiple tables and multiple result tables at the same time.

We suggest a design for implementing LIKE expressions in viSQLizer. The LIKE expression could go through (and hover) the different cells in the original table to find a match. Once a match is found, the cell is animated from the original table to the result table and the search continues for more matches. By visualizing LIKE expressions this way, students might understand why certain cells fulfill the LIKE expressions' criteria and thus are animated into the result table.

Table operations could also be implemented in the future. To prevent the database data to be altered or deleted by the students, a copy could be created for each individual student. This approach was done by Grillenberger & Brina [24]. The visualization of the different table operations might depend on the type of operation; CREATE TABLE, ALTER TABLE, INSERT/UPDATE/DELETE. A database model could be displayed to the student, or a table could appear/disappear in the visualization.

## References

- [1] Atchison, W. F., Conte, S. D., Hamblen, J. W., Hull, T. E., Keenan, T. A., Kehl, W. B., McCluskey, E. J., Navarro, S. O., Rheinboldt, W. C., Schweppe, E. J., Viavant, W., & Young, Jr., D. M. March 1968. Curriculum 68: Recommendations for academic programs in computer science: A report of the acm curriculum committee on computer science. *Commun. ACM*, 11(3), 151–197.
- [2] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA, 999133.
- [3] Mitrovic, A. 1998. A knowledge-based teaching system for sql. In *Proceedings of ED-MEDIA*, volume 98, 1027–1032.
- [4] Kearns, R., Shead, S., & Fekete, A. 1996. A teaching system for sql. In *Proceedings of the 2Nd Australasian Conference on Computer Science Education, ACSE '97*, 224–231, New York, NY, USA. ACM.
- [5] Hansen, K. T. & Mallaug, T. 2008. *Databaser*. Gyldendal Norsk Forlag - Gyldendal akademisk, 2 edition.
- [6] Sadiq, S., Orlowska, M., Sadiq, W., & Lin, J. 2004. Sqlator: An online sql learning workbench. *SIGCSE Bull.*, 36(3), 223–227.
- [7] Cembalo, M., De Santis, A., & Ferraro Petrillo, U. 2011. Savi: A new system for advanced sql visualization. In *Proceedings of the 2011 Conference on Information Technology Education, SIGITE '11*, 165–170, New York, NY, USA. ACM.
- [8] Prior, J. C. 2003. Online assessment of sql query formulation skills. In *Proceedings of the Fifth Australasian Conference on Computing Education - Volume 20, ACE '03*, 247–256, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

- [9] Folland, K. A. T. visqlizer: An interactive visualizer for learning sql. Master's thesis, NTNU Gjøvik, Norway, 2016.
- [10] VertabeloAcademy. Sql queries. <https://academy.vertabelo.com/course/sql-queries>. (Visited Feb. 2016).
- [11] Cumming, A. Sqlzoo. <http://sqlzoo.net>. (Visited May 2016).
- [12] CodeAcademy. Learn sql. <https://www.codecademy.com/learn/learn-sql>. (Visited Feb. 2016).
- [13] Means, B., Toyama, Y., Murphy, R., Bakia, M., & Jones, K. 2009. Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies. *US Department of Education*.
- [14] GalaXQL. Galaxql 3.0. <http://sol.gfxile.net/g3/>. (Visited Feb. 2016).
- [15] Schemaverse. The schemaverse. <https://schemaverse.com>. (Visited Feb. 2016).
- [16] Belov, A. Serious game for learning relational databases. Master's thesis, Gjøvik University College, Norway, 2004.
- [17] KhanAcademy. Intro to sql: Querying and managing data. <https://www.khanacademy.org/computing/computer-programming/sql>. (Visited Feb. 2016).
- [18] Murray, M. & Guimaraes, M. 2008. Animated database courseware: Using animations to extend conceptual understanding of database concepts. *J. Comput. Sci. Coll.*, 24(2), 144–150.
- [19] Allenstein, B., Yost, A., Wagner, P., & Morrison, J. 2008. A query simulation system to illustrate database query execution. *SIGCSE Bull.*, 40(1), 493–497.
- [20] Patwardhan, M. & Murthy, S. 2015. When does higher degree of interaction lead to higher learning in visualizations? exploring the role of 'interactivity enriching features'. *Computers & Education*, 82, 292 – 305.
- [21] Danaparamita, J. & Gatterbauer, W. 2011. Queryviz: Helping users understand sql queries and their patterns. In *Proceedings of the 14th International Conference on Extending Database Technology, EDBT/ICDT '11*, 558–561, New York, NY, USA. ACM.
- [22] Swanhart, J. & Rothe, A. Php-sql-parser. <https://github.com/greenlion/PHP-SQL-Parser>. (Visited Feb. 2016).
- [23] Brovold, M., Johnsen, F., & Folland, K. A. T. 2014. Decomposer. <https://dev.imt.hig.no/confluence/display/SQLD/SQL+Decomposer+Home>. (Visited Apr. 2016).
- [24] Grillenberger, A. & Brinda, T. 2012. eledsql: A new web-based learning environment for teaching databases and sql at secondary school level. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education, WiPSCE '12*, 101–104, New York, NY, USA. ACM.